UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/812,373	03/29/2004	Xiaohua Shi	ITL.1096US (P18491)	7604

21906   7590   11/16/2007
TROP PRUNER & HU, PC
1616 S. VOSS ROAD, SUITE 750
HOUSTON, TX 77057-2631

EXAMINER
WANG, JUE S

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
11/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/812,373	SHI ET AL.
	Examiner	Art Unit	
	Jue S. Wang	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>06 September 2007</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-30</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-30</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>29 March 2004</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

1.   Claims 1-30 have been examined.

Claim Rejections - 35 USC § 112

2.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3.   Claims 13-15 and 27-29 are rejected under 35 U.S.C. 112, second paragraph, as being

indefinite for failing to particularly point out and distinctly claim the subject matter which

applicant regards as the invention.

A. The following claim language is not clear and indefinite:

i.   As per claim 13, lines 2-3, and claim 27, line 2, the phrase "m rows and m

columns" is used. It is not clearly understood what is meant by this limitation (i.e.,

does "m rows and m columns" mean that the scoreboard is extended by an

arbitrary number of rows and columns, or the scoreboard is extended by an

arbitrary but equal number of rows and columns, or the scoreboard is extended by

a number of rows and columns that is related to the number of instructions being

scheduled, etc.).

ii.   As per claim 14, line 2, and claim 28, line 2, the phrase "an m-th row" is

used. It is not clearly understood what is meant by this limitation (i.e., does "m-th

row" refer to any row in the scoreboard, to the last row in the register scoreboard

with m-rows and m-columns, or is "m-th" row a specific row that must be used in

relation to the instruction that is reordered, etc.).

iii.   As per claim 15, line 2, and claim 29, line 2, the phrase "an m-th column"

is used. It is not clearly understood what is meant by this limitation (i.e., does "m-

th column" refer to any column in the scoreboard, to the last column in the

register scoreboard with m-rows and m-columns, or is "m-th" row a specific row

that must be used in relation to the instruction that is reordered, etc.).

Appropriate corrections are required.

Any claim not specifically addressed, above, is being rejected as incorporating the

deficiencies of a claim upon which it depends.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 1, 2, 16, 21, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Wang et al. (US 7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining:

An Effective Scheduling Technique for VLIW Machines".

6.   As per claim 1, Wang teaches the invention as claimed including a method comprising:

assigning a number of stall cycles between a first and a second instruction (see column

30, lines 4-7, 22-27, 43-48, 55-60, and column 34, line 50-column 35, line 3; EN: while Wang

states determining a number of stall cycles between a pair of instructions, the determining

incorporates assigning the stall cycles resulting from the determination to the pair of instructions

since the subsequent scheduling of this pair instructions is based on the number of stall cycles to

minimize the total stall cycles which is the same goal of the applicants' invention to "eliminate

most of GAPs in many Java applications" as disclosed in page 6, line 3-page 7, line 2 of the

specification); and

scheduling said first and second instructions for execution based on the assigned stall

cycles (see column 30, lines 7-10, and column 35, lines 12-19).

Wang does not teach determining a range within which one of said instructions may be

reordered without violating data dependency and reordering said instruction within said range.

Lam teaches a scheduling technique for software piplelining, including determining a

range within which one minimally indivisible sequences of micro-instructions may be reordered

without violating data dependency and reordering the sequence of instructions within said range

(see page 320, section 2.1, page 322, left column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang to determine a range within which one of said instructions may be

reordered without violating data dependency and reordering said instruction within said range as

taught by Lam because violating data dependency such as scheduling the use of a result before it

is ready will result in pipeline stalls (see column 34, lines 53-56 of Wang, page 320, right

column of Lam) which would interfere with the goal of instruction scheduling which is to

minimize pipeline stalls (see column 30, lines 7-10 of Wang).

7.   As per claim 2, Wang teaches using a number of the maximum possible pipeline stall

cycles between said first and second instructions to indicate a data dependency therebetween (see

column 30, lines 12-26).

8.   As per claim 16, this is a system claim of claim 1. Therefore, it is rejected for the same

reason as claim 1.

9.   As per claims 21-22, they are the article comprising a computer readable storage medium

for storing instruction claims of claims 1 and 2. Therefore, they are rejected for the same reasons

as claims 1 and 2.

10.   Claims 3, 17-19, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Wang et al. (US 7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining: An

Effective Scheduling Technique for VLIW Machines", as applied to claims 2, 16, and 22 above,

further in view of Larson et al., (US 6,662,293 B1, hereinafter Larson).

11.   As per claim 3, Wang and Lam do not teach extending a register scoreboard that keeps

track of the data dependency.

        Larson teaches extending a register scoreboard that keeps track of the data dependency

(see column 1, line 67 and column 2, lines 1-18).

        It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang and Lam with the feature of extending a register scoreboard that keeps

track of the data dependency as taught by Larson because processors that support out-of-order

execution often use an "instruction scoreboard" to keep track of information regarding

dependencies between instructions (see column 1, lines 31-35 of Larson).

12.   As per claim 17, Wang and Lam do not teach a register to store dependency data between

said first and second instruction.

        Larson teaches a register to store dependency data between a first and second instruction

(see column 1, line 67 and column 2, lines 1-18).

        It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang and Lam with the feature of a register scoreboard to store dependency

data between said first and second instruction as taught by Larson because processors that

support out-of-order execution often use an "instruction scoreboard" to keep track of information

regarding dependencies between instructions (see column 1, lines 31-35 of Larson).

13.   As per claim 18, Wang further teaches a compiler coupled to schedule said first and

second instructions for execution based on a maximum possible pipeline stall cycles (see column

30, lines 4-26).

14.   As per claim 19, Larson further teaches that the register is a register scoreboard (see

column 2, lines 1-18)..

15.   As per claim 23, this is the article comprising a computer readable storage medium for

storing instruction claim of claim 3. Therefore, it is rejected for the same reason as claim 3.

16.   Claims 4 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US 7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining: An Effective Scheduling Technique for VLIW Machines", further in view of Larson et al., (US 6,662,293 B1, hereinafter Larson), as applied to claims 3 and 23 above, further in view of Anderson et al. (US 6,092,180, hereinafter Anderson).

17.   As per claim 4, Wang, Lam and Larson do not teach maintaining a count of issue latency for said first and second instructions.

Anderson teaches maintaining a count of issue latency for the sampled instructions (see column 7, lines 7-13 and column 16, lines 19-23).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Wang as modified to maintain a count of issue latency for said first and second instructions as taught by Anderson because many of the conditions associated with stalls are difficult to predict at compile-time, and all of them degrade the performance of the system, so it is valuable to sample the information (see column 11, lines 31-40 of Anderson).

18.   As per claim 24, this is the article comprising a computer readable storage medium for storing instruction claim of claim 4. Therefore, it is rejected for the same reason as claim 4.

19.   Claims 5, 6, 10, 25, 26, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US 7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining: An Effective Scheduling Technique for VLIW Machines", further in view of Larson et al., (US

6,662,293 B1, hereinafter Larson), as applied to claims 3 and 23 above, further in view of Levine

et al. (US 5,987,598, hereinafter Levine).

20.   As per claim 5, Wang, Lam and Larson do not teach maintaining a count for the number

of cycles from start to end of an issue of said first and second instructions.

Levine teaches maintaining a count for the number of cycles from start to end of an issue

of said first and second instructions (see column 6, lines 29-31).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang as modified to maintain a count for pipeline stalls between said first

instruction and a previous instruction as taught by Levine because the performance monitor

permit identifying performance bottlenecks and monitoring pipeline stalls, so that software

engineer can readily determine what modifications to the code can be made to optimize data

processing system performance (see column 4, lines 49-63 and column 7, lines 40-52 of Levine),

so in particular, it would have been obvious that compiler writers can use the performance data

to evaluate the effectiveness of a scheduling algorithm.

21.   As per claim 6, Wang and Larson do not teach maintaining a count for pipeline stalls

between said first instruction and a previous instruction.

Levine teaches maintaining a count for pipeline stalls between said first instruction and a

previous instruction (see column 5, lines 1-4, 53-61, and column 6, lines 33-40).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang as modified to maintain a count for pipeline stalls between said first

instruction and a previous instruction as taught by Levine because the performance monitor

permit identifying performance bottlenecks and monitoring pipeline stalls, so that software engineer can readily determine what modifications to the code can be made to optimize data processing system performance (see column 4, lines 49-63 and column 7, lines 40-52 of Levine), so in particular, it would have been obvious that compiler writers can use the performance data to evaluate the effectiveness of a scheduling algorithm.

22.   As per claim 10, Wang, Lam and Larson do not teach keeping track of an instruction that causes pipeline stall.

Levine teaches keeping track of an instruction that causes pipeline stall (see column 5, lines 46-61).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Wang as modified to keep track of an instruction that causes pipeline stall as taught by Levine because the performance monitor permit identifying performance bottlenecks and monitoring pipeline stalls, so that software engineer can readily determine what modifications to the code can be made to optimize data processing system performance (see column 4, lines 49-63 and column 7, lines 40-52 of Levine), so in particular, it would have been obvious that compiler writers can use the performance data to evaluate the effectiveness of the scheduling algorithm.

23.   As per claims 25, 26, and 30, they are the article comprising a computer readable storage medium for storing instruction claims of claims 5, 6, and 10. Therefore, they are rejected for the same reasons as claims 5, 6, and 10.

24.   Claims 7-9 and 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Wang et al. (US 7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining: An

Effective Scheduling Technique for VLIW Machines", further in view of Larson et al., (US

6,662,293 B1, hereinafter Larson), as applied to claims 3 and 23 above, further in view of

Merchant et al. (US 6,334,182 B2, hereinafter Merchant).

25.   As per claim 7, Larson teaches the register scoreboard keeps track of data dependency

(see Fig 3, column 5, lines 40-62). However, Wang, Lam, and Larson do not teach that the

register scoreboard keep track of a maximum possible pipeline stall cycles.

Merchant teaches a dependency matrix with m-rows and m-columns (see column Fig 5,

Fig 6, column 4, lines 43-57, and column 5, lines 42-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang modified by Larson such that the register scoreboard is extended by m-

rows and m-columns as taught by Merchant because by using a dependency matrix, the

dependencies between entries in the scheduling queue are easily setup and maintained, and the

time consuming compares in the scheduling loop may be eliminated (see column 4, lines 36-42

of Merchant). Furthermore, while Merchant only teaches that a dependency between two

instructions in the dependency matrix is represented as "D" which could be a bit having a value

of "1" (see column 4, lines 43-50 of Merchant), it would have been obvious to one of ordinary

skill in the art that "D" could be represented by the maximum possible pipeline stall cycles since

Wang uses a number of the maximum possible pipeline stall cycles between said first and second

instructions to indicate a data dependency therebetween (see column 30, lines 12-26 of Wang)

and using the number of stall cycles provides information about the extend of the dependency

between two instructions rather than just the existence of the dependency.

26.   As per claim 8, Merchant further teaches keeping track of a first non-zero value from

right to left in m-th row of the dependency matrix to reorder said first instruction (see column 6,

lines 53-65 and column 7, lines 20-31; EN: the ready determination unit keeps track of a first

non-zero value in the m-row since an entry with a bit set with a value of "1" indicates that the

instruction is not ready to be dispatched. While Merchant does not disclose the direction of the

ready determination unit when checking the row, it would have been obvious to one of ordinary

skill in the art at the time of the invention that the ready determination unit could operate in the

right to left direction since any non-zero value found in the row regardless of the operation it

corresponds to will indicate that the instruction associated with the row is not ready to dispatch).

27.   As per claim 9, Merchant further teaches keeping track of a first non-zero value from top

to bottom in m-th column of the dependency matrix to reorder said first instruction (see column

6, lines 62-65 and column 7, lines 28-31; EN: the ready determination unit keeps track of a first

non-zero value in the m-th column by making sure to clear the non-zero value when the

operation associated with the row is dispatched, which in turns allows the scheduling of

instructions dependent on the dispatched instruction. While Merchant does not disclose the

direction in which the column is cleared, it would have been obvious to one of ordinary skill in

the art that the column can be cleared in a top to bottom order since the column is cleared

regardless of the direction of the clearing process).

28.   As per claims 27-29, they are the article comprising a computer readable storage medium

for storing instruction claims of claims 7-9. Therefore, they are rejected for the same reasons as

claims 7-9.

29.   Claims 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang

et al. (US 7,036,106 B1, hereinafter Wang), in view of Anderson et al. (US 6,092,180,

hereinafter Anderson), further in view of Lam, "Software Pipelining: An Effective Scheduling

Technique for VLIW Machines".

30.   As per claim 11, Wang teaches a compiler that schedules a first and second instructions

for execution based on the stall cycles between the first and second instruction (see column 30,

lines 4-10 and column 35, lines 12-19).

Wang does not teach that the compiler is coupled to an apparatus with a register to store a

number of stall cycles between a first and a second instruction.

Anderson teaches an apparatus with a register to store a number of stall cycles between a

first instruction and a second instruction (see Fig 3, column 7, lines 7-13 and column 12, lines

57-63).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang to couple the compiler with an apparatus with a register to store a

number of stall cycles between a first and a second instruction as taught by Anderson because

static instruction schedulers do not always have exact dependency information which would

allow them to optimally schedule the memory access instructions, and it is difficult to exactly

predict the latency of memory access instructions (see column 27, lines 9-29 of Anderson).

Wang and Anderson do not teach that the compiler determines a range within which one

of said instructions may be reordered without violating data dependency and reordering said

instruction within said range.

Lam teaches a scheduling technique for software piplelining, including determining a

range within which one minimally indivisible sequences of micro-instructions may be reordered

without violating data dependency and reordering the sequence of instructions within said range

(see page 320, section 2.1, page 322, left column, paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang and Anderson to determine a range within which one of said instructions

may be reordered without violating data dependency and reordering said instruction within said

range as taught by Lam because violating data dependency such as scheduling the use of a result

before it is ready will result in pipeline stalls (see column 34, lines 53-56 of Wang, page 320,

right column of Lam) which would interfere with the goal of instruction scheduling which is to

minimize pipeline stalls (see column 30, lines 7-10 of Wang).

31.   As per claim 12, Wang further teaches that the compiler uses a number of maximum

possible pipeline stall cycles between said first and second instructions to indicate data

dependency therebetween (see column 30, lines 12-26).

32.   Claims 13-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al.

(US 7,036,106 B1, hereinafter Wang), in view of Anderson et al. (US 6,092,180, hereinafter

Anderson), further in view of Lam, "Software Pipelining: An Effective Scheduling Technique

for VLIW Machines", as applied to claim 12 above, further in view of Merchant et al. (US

6,334,182 B2, hereinafter Merchant).

33.   As per claim 13, Wang and Anderson do not teach that the register is extended by m-

rows and m-columns to keep track of maximum possible pipeline stall cycles.

Merchant teaches a dependency matrix with m-rows and m-columns (see column Fig 5,

Fig 6, column 4, lines 43-57, and column 5, lines 42-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to have modified Wang modified by Larson such that the register scoreboard is extended by m-

rows and m-columns as taught by Merchant because by using a dependency matrix, the

dependencies between entries in the scheduling queue are easily setup and maintained, and the

time consuming compares in the scheduling loop may be eliminated (see column 4, lines 36-42

of Merchant). Furthermore, while Merchant only teaches that a dependency between two

instructions in the dependency matrix is represented as "D" which could be a bit having a value

of "1" (see column 4, lines 43-50 of Merchant), it would have been obvious to one of ordinary

skill in the art that "D" could be represented by the maximum possible pipeline stall cycles since

Wang uses a number of the maximum possible pipeline stall cycles between said first and second

instructions to indicate a data dependency therebetween (see column 30, lines 12-26 of Wang)

and using the number of stall cycles provides information about the extend of the dependency

between two instructions rather than just the existence of the dependency.

34.   As per claim 14, Merchant further teaches keeping track of a first non-zero value from

right to left in m-th row to reorder said first instruction (see column 6, lines 53-65 and column 7,

lines 20-31; EN: the ready determination unit keeps track of a first non-zero value in the m-row

since an entry with a bit set with a value of "1" indicates that the instruction is not ready to be

dispatched. While Merchant does not disclose the direction of the ready determination unit when

checking the row, it is obvious to one of ordinary skill in the art that the ready determination unit

can operate from right to left direction since any non-zero value found in the row regardless of

the operation it corresponds to will indicate that the instruction associated with the row is not

ready to dispatch). Merchant does not disclose the use of the dependency matrix by a compiler,

however, it would have been obvious to one of ordinary skill in the art that the idea and principle

of the dependency matrix could be used in a compiler which must keep track of dependencies

between instructions when scheduling instructions.

35.   As per claim 15, Merchant further teaches keeping track of a first non-zero value from

top to bottom in m-th column to reorder said first instruction (see column 6, lines 62-65 and

column 7, lines 28-31; EN: the ready determination unit keeps track of a first non-zero value in

the m-th column by making sure to clear the non-zero value when the operation associated with

the row is dispatched, which in turns allows the scheduling of instructions dependent on the

dispatched instruction. While Merchant does not disclose the direction in which the column is

cleared, it would have been obvious to one of ordinary skill in the art that the column can be

cleared in a top to bottom order since the column is cleared regardless of the direction of the

clearing process). Merchant does not disclose the use of the dependency matrix by a compiler,

however, it would have been obvious to one of ordinary skill in the art that the idea and principle

of the dependency matrix could be used in a compiler which must keep track of dependencies

between instructions when scheduling instructions.

36.   Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US

7,036,106 B1, hereinafter Wang), in view of Lam, "Software Pipelining: An Effective

Scheduling Technique for VLIW Machines", further in view of Larson et al., (US 6,662,293 B1,

hereinafter Larson), as applied to claim 17 above, further in view of Cyran et al. (US 6,412,107

B1, hereinafter Cyran).

37.   As per claim 20, Wang, Lam and Larson do not teach that the compiler is just-in-time

compiler for an object-oriented programming language.

Cyran teaches a just-in-time compiler for an object oriented programming language (see

column 2, lines 42-65 and column 5, lines 65-67).

It would have been obvious to one of ordinary skill in the art at the time of the invention

to use the specific instruction scheduling technique taught by Wang, Lam, and Larson in a just-

in-time compiler for an object oriented programming language as in Cyran because instruction

scheduling algorithms reorder instructions to increase performance by reducing pipeline stalls

and these algorithms operate by simulating the target processor's pipeline to determine the

instruction ordering that results in the fewest number of stall cycles (see column 29, line 63 -

column 30, lines 3 of Wang).

Response to Arguments

38.   Rejection of Claims 4, 5, 7-9, 13-15, 17-20, 22-30 rejected under 35 U.S.C. 112, second

paragraph:

39.   As per claims 4, 5, 17, 24, and 25, applicants' arguments have been fully considered and

are persuasive. Therefore, the rejections of claims 4, 5, 17, 24, and 25 under 35 U.S.C. 112,

second paragraph have been withdrawn.

40.   The rejections of claims 7-9 are overcome by applicants' amendments. Claims 13-15 and

27-29 are not amended and contain similar indefinite claim language as the original claims 7-9

before amendment, therefore, the rejections of these claims under 35 U.S.C. 112, second

paragraph are maintained.

41.   <u>Rejection of Claims 1, 11, 16, and 21 under 35 U.S.C. § 102(e)</u>:

42.   Applicants' arguments with respect to claims 1, 11, 16, and 21 have been considered but

are moot in view of the new ground(s) of rejection.

Conclusion

43.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

o   Rober Walker and Samit Chaudhuri, "Introduction to the Scheduling Problem", 1995,

    Design & Test of Computers, Volume 12, Issue 2.

    This document is cited to teach the use of a schedule interval in scheduling operations

    where the length of the interval is the mobility of the operation.

44.   Applicant's amendment necessitated the new ground(s) of rejection presented in this

office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

45.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The

examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jue Wang
Examiner
Art Unit 2193

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100